Adaptive Resource Management Technology for Onboard Satellite Systems

Lonnie Welch, David Chelberg

Ohio University

Presented By

David Fleeman

Ohio University

NASA Earth Science Techology Conference

June 24, 2003

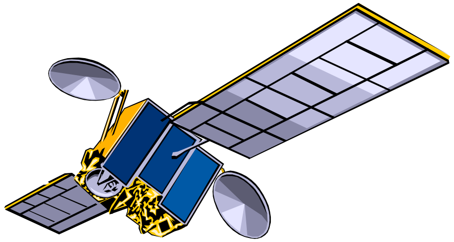
Contents

- Motivation
- Unique Issues for Satellite Systems
  - Space-to-Ground Downlink
  - Satellite Buffer Management
- Resource Management Architecture
  - Components
  - System Parameters for Performance Adaptability
  - Benefit Model
- Example Satellite System
- Future Work

Definitions

- Adaptive Resource Management
  - Autonomous control of the resource usage of applications such that:
    - real-time constraints are satisfied
    - the benefit of captured data is optimal (i.e., quality of service (QoS) is optimized)
- Benefit
  - The scientific value of captured data as determined by scientists or application system designers.

Context



- Earth-observing satellites operating within any of the following contexts:
  - Sensors dynamically gather more data than can be processed in real-time.
  - Sensors dynamically gather more data than can be stored or downloaded to the Earth.
  - Multiple on-board systems having dynamic resource needs and compete for resources.
- Other dynamic contexts
  - Sensor Web
  - Ground Systems

Identified Problems With Management of Current Satellite Systems

- Operations costs are high
  - Satellites need full-time ground support to monitor and maintain satellite health
  - Flight Operations Teams are needed to solve resource allocation anomalies

- Benefit of collected data is sub-optimal
  - Valuable data may be discarded when storage is full
  - Anomalies may persist for prolonged periods of time due to:
    - Downlink latencies
    - Missed downlink opportunities
    - Offline monitoring and maintenance

- Limited capabilities available for graceful degradation in overload due to these anomalies

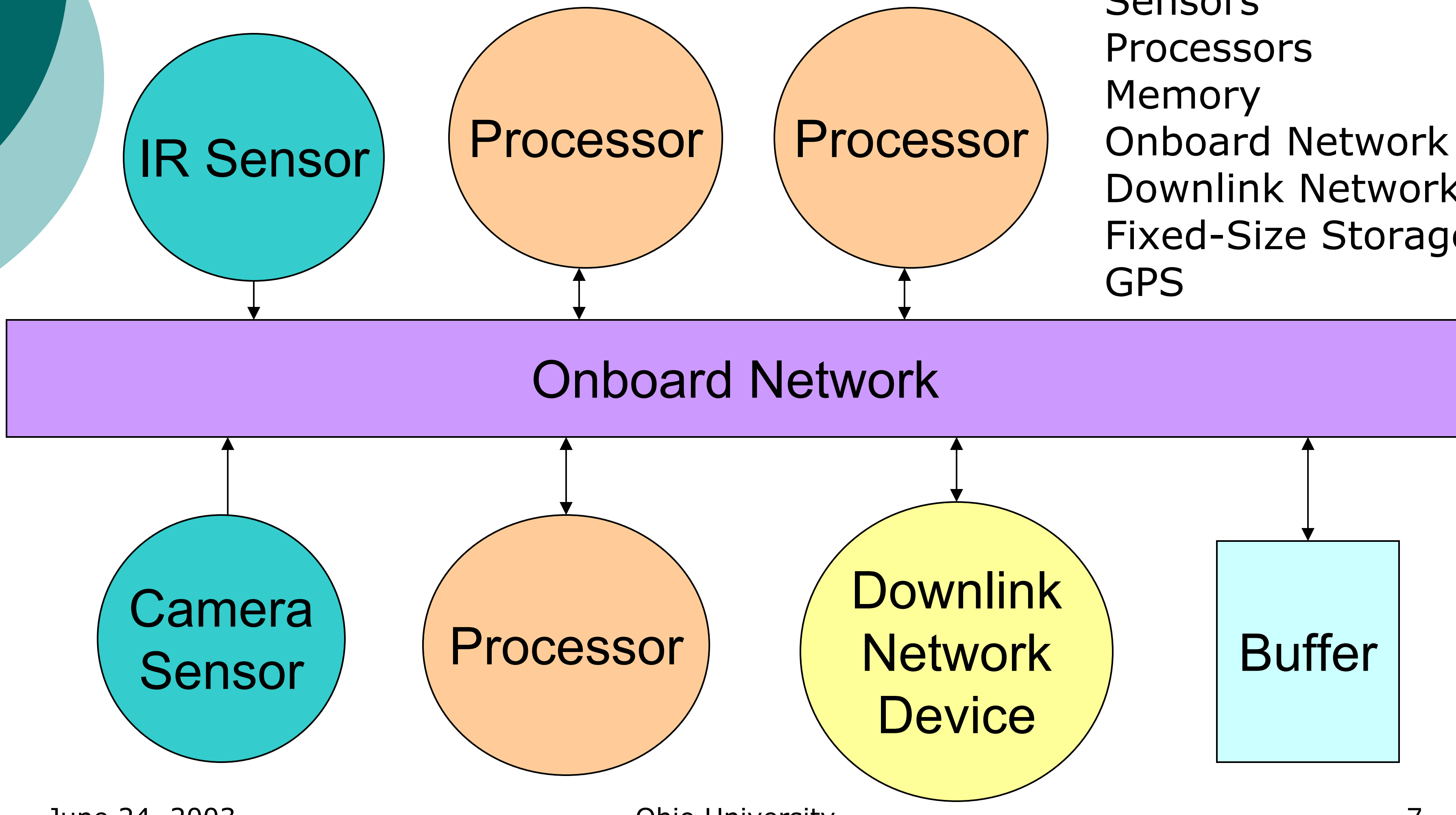
Proposed Solution
Adaptive Resource Management

- Satellite control mechanisms autonomously optimize the performance of onboard systems using the following mechanisms:
  - Tasks are dynamically assigned to resources.
  - Systems operate at various levels of fidelity allowing resource usage to vary.
  - Systems produce varying amounts of data depending on resource availability and environmental conditions.
  - Stored data of the greatest benefit is downloaded first.
  - Additional control mechanisms can be identified.

- We have developed a middleware solution for a simulated Earth-observing satellite system.

Hardware Components

Sensors
Processors
Memory
Onboard Network
Downlink Network
Fixed-Size Storage
GPS



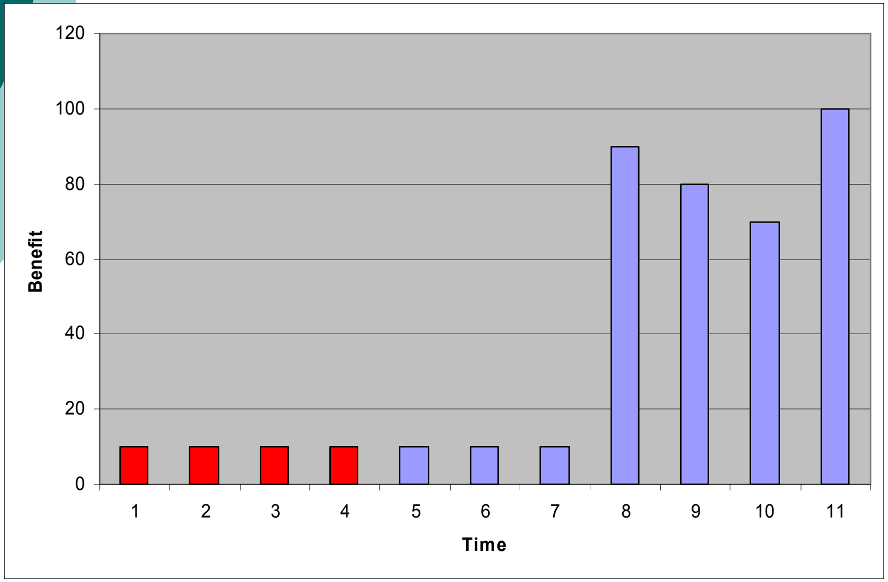
Downlink Network Device

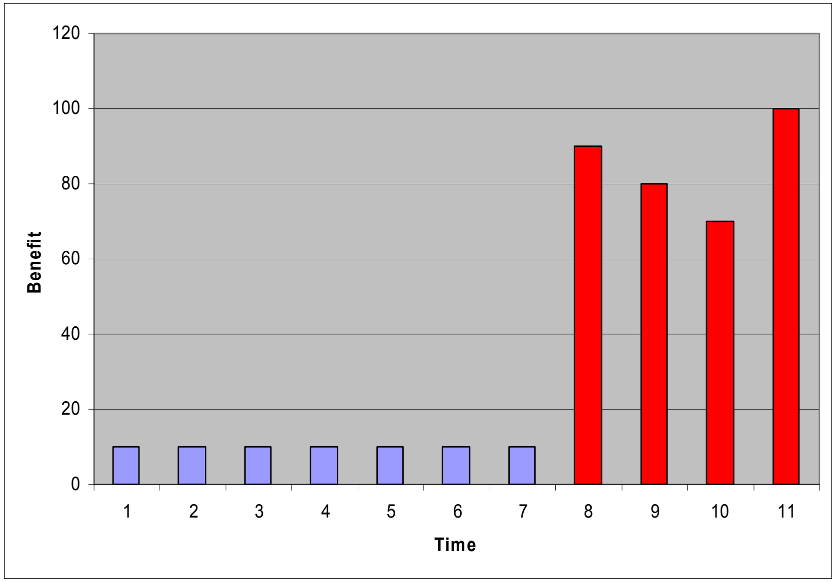
- Assumptions:
  - Network downlink is not always available.
  - A downlink schedule containing contact times and expected bandwidths is known *a priori.*
  - Downlink anomalies occur due to unforeseen events such as missed downlink opportunities caused by poor weather conditions or other types of failure.
- Resource Management must make onboard adjustments to avoid losing scientific data when overload situations occur due to downlink anomalies.

Satellite Buffer Management

- Assumptions:
  - Buffer is random-access such that all data is stored in files that can be accessed at any time.
  - Buffer may become full due to excess data collection by satellite applications or downlink anomalies, requiring scientific data to be discarded.
- Resource Management can avoid losing valuable scientific data by preventing buffer overload situations to occur by controlling the resource needs of satellite applications and by recovering from downlink anomalies.

Buffer Management in Overload





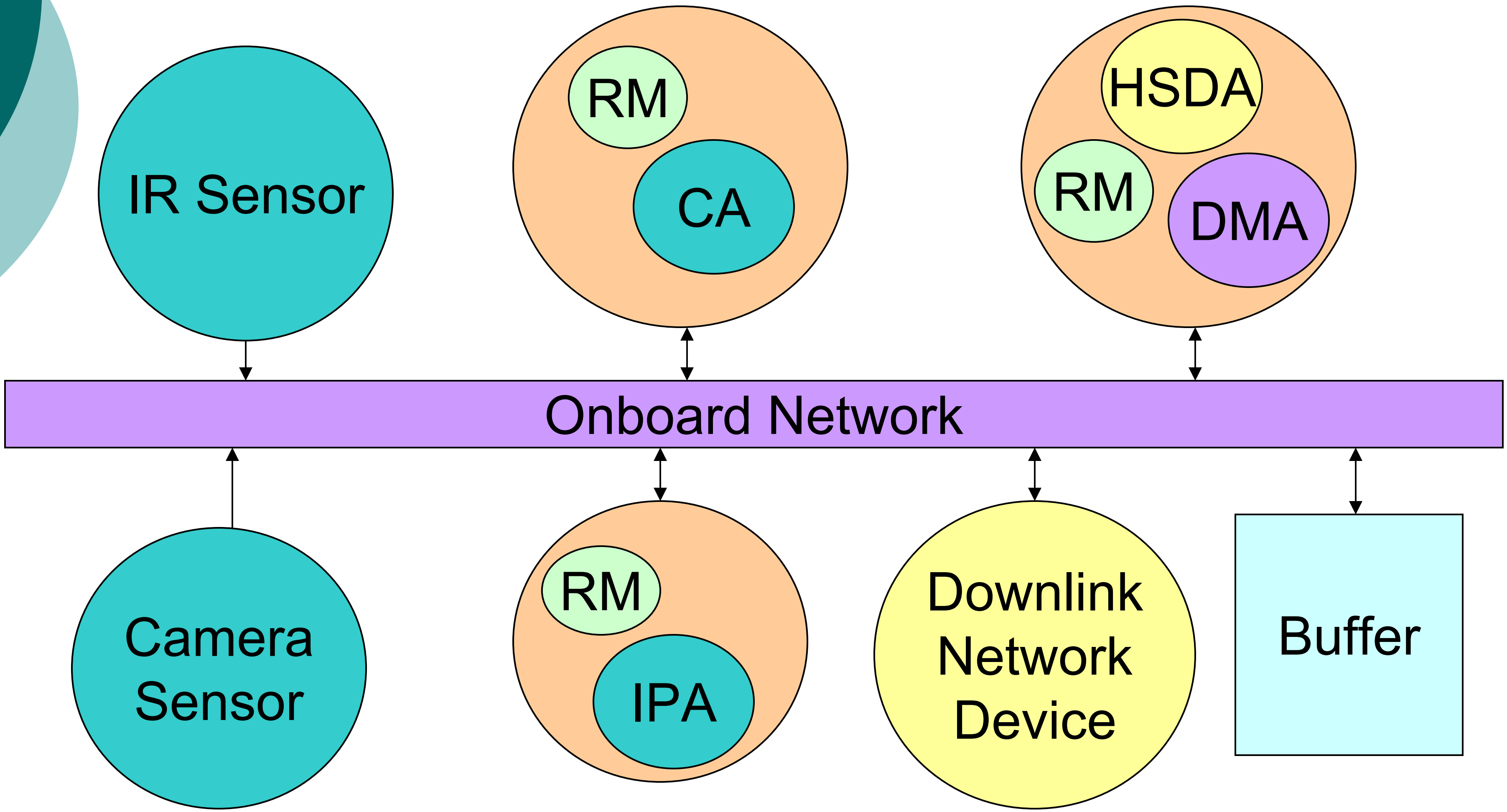
The buffer on these satellites has room for only four images. The graph on the left shows what is downloaded (in red) in the present satellite model. The right shows what would be downloaded in the proposed satellite model.

Proposed Software Components of Satellites of the Near Future

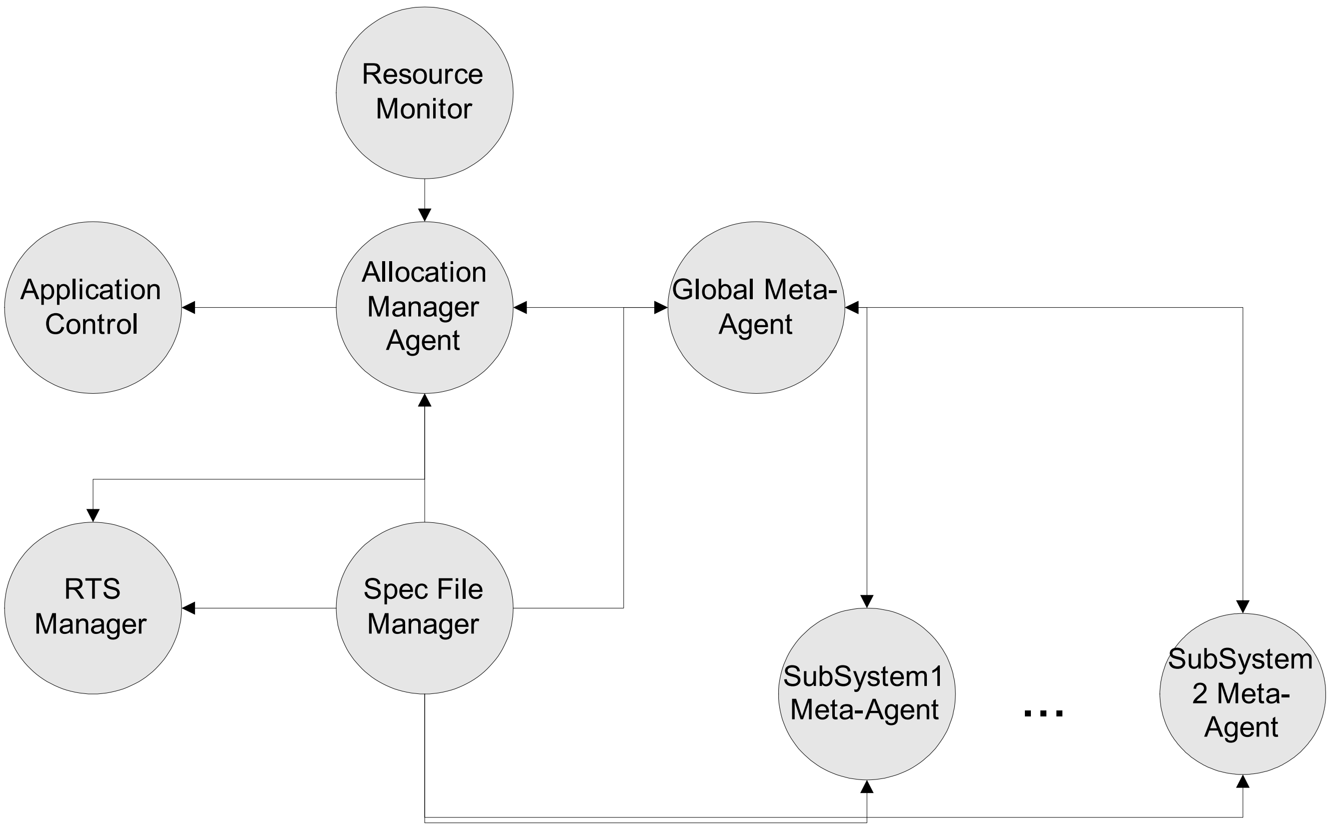
- High level components
  - Operating System
  - Resource Management Middleware
  - Satellite Monitoring Software
  - Science Processing Software

- Example satellite applications
  - Health and Safety Data Agent (HSDA)
  - Image Processing Agent (IPA)
  - Compression Agent (CA)
  - Data Management Agent (DMA)

Example Mapping of Software Components To Hardware



Resource Management Middleware



Modeling System Parameters

- Application performance can be affected by two types of conditions:
  - Environmental conditions
  - Quality of Service parameters
- **Extrinsic Attribute** – a name-value pair that describes the condition of some element in the environment that may affect the resource usage or benefit of an application. (i.e., cloud-cover)
- **Service Attribute** – a name-value pair that describes the quality of service provided by some dimension of an application. (i.e., number of pixels to process)

Software Specification

```
Extrinsic Attributes {
    Attribute temperature {
      DOUBLE CONTINUOUS [0.0, 10000.0];  // Kelvin
      Default [300.0];
    }
    Attribute power {
      DOUBLE CONTINUOUS [0.0, 10000.0];  // Watts
      Default [9000.0];
    }
    Attribute pressure {
      DOUBLE CONTINUOUS [0.0, 3.0]; //atmospheres
      Default [1.0];
    }
}
```

Software Specification

```
Service Attributes {
   Attribute service_temperature {
     STRING [NONE, LOW, MEDIUM, HIGH];
     Default [LOW];
     Affects [hsda];
   }
   Attribute service_power {
     STRING [NONE, LOW, MEDIUM, HIGH];
     Default [LOW];
     Affects [hsda];
   }
   Attribute service_pressure {
     STRING [NONE, LOW, MEDIUM, HIGH];
     Default [LOW];
     Affects [hsda];
   }
}
```

Software Specification

```
Tables {
  Benefit Table HSDA (Set service_power, service_pressure,
                                                service_temperature) {
    Header power, pressure, temperature, service_power,
                 service_pressure, service_temperature, benefit;
    Row    10000.0,  3.0,  10000.0,  HIGH,    HIGH,    HIGH,     1.0;
    Row      100.0,  0.0,      0.0,  HIGH,    HIGH,    HIGH,     1.0;
    Row     5000.0,  1.5,   5000.0,  HIGH,    HIGH,    HIGH,     1.0;
    Row    10000.0,  3.0,  10000.0,  NONE,    NONE,    NONE,     0.0;
    Row      100.0,  0.0,      0.0,  NONE,    NONE,    NONE,     0.0;
    Row     5000.0,  1.5,   5000.0,  NONE,    NONE,    NONE,     0.0;
    Row    10000.0,  3.0,  10000.0,  LOW,     LOW,     LOW,      0.1;
    Row      100.0,  0.0,      0.0,  LOW,     LOW,     LOW,      0.1;
    Row     5000.0,  1.5,   5000.0,  LOW,     LOW,     LOW,      0.3;
    Row    10000.0,  3.0,  10000.0,  MEDIUM,  MEDIUM,  MEDIUM,   0.6;
    Row      100.0,  0.0,      0.0,  MEDIUM,  MEDIUM,  MEDIUM,   0.6;
    Row     5000.0,  1.5,   5000.0,  MEDIUM,  MEDIUM,  MEDIUM,   0.9;
    Row    10000.0,  3.0,  10000.0,  HIGH,    LOW,     LOW,      0.3;
    Row      100.0,  0.0,      0.0,  LOW,     LOW,     HIGH,     0.3;
    Row     5000.0,  1.5,   5000.0,  HIGH,    HIGH,    MEDIUM,  0.98;
  }
}
```

Decisions Made to Achieve Performance Requirement

- Allocation of applications to processors
- Choose service attribute values which affects the benefit to the scientist and can affect the usage of all the following resources:
  - CPU
  - Memory
  - Network
  - Buffer
- Optimize the benefit of the captured data locally by adjusting service attribute values in view of observation schedules and perturbations in the environment (i.e., *extrinsic attributes*).

Allocation Manager

- Has complete knowledge of all the real-time systems and resources.
- Proactively and reactively maintains the allocation (mapping of applications to resources) such that real-time requirements are satisfied and benefit is maximized.
- In both the following cases, the Allocation Manager may decide to change the allocation in order to improve the quality of service for the complete system.
  - Receives events from RTS Manager when latency violations are detected or predicted.
  - Receives events from the Global Meta-Agent when it determines a change in the system allocation is either necessary or may be beneficial.

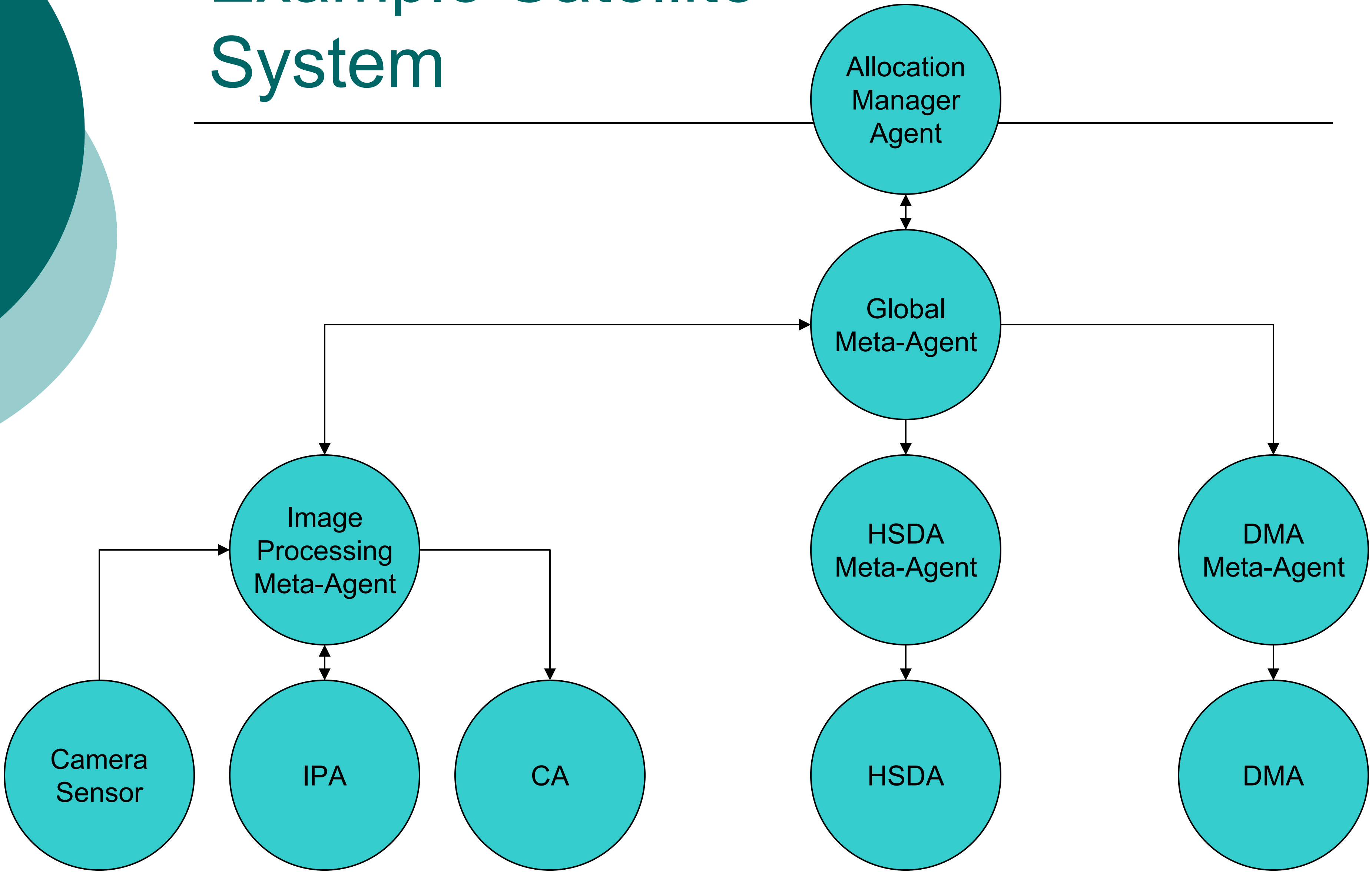
Global Meta-Agent

- Determines when current system settings are no longer optimal or feasible.
- Determines service attribute settings that optimize benefit for the entire system, and then sends the corresponding settings to the Meta-Agent for each subsystem.
- Negotiates with Meta-Agents to achieve system-level, optimal allocations.
- Triggers reallocations when an allocation becomes sub-optimal.
- Triggers service attribute changes when system level benefit becomes sub-optimal.

Meta-Agent

- Each Meta-Agent represents a sub-system and is responsible for monitoring the extrinsic attributes and setting the service attributes for each agent.
- Meta-Agents perform a local maximization over the service attributes associated with their related sub-systems. If they find a better setting of service attributes for their sub-system, they request it from the Global Meta-Agent.

Example Satellite System



Camera Sensor / Simulator

- The Camera Sensor (CS) will make a new image available to the IPA.
- The CS will send unprocessed images to the buffer for processing by other agents.
- The frequency of collecting images and the importance of an observation can either be given *a priori* in an observation schedule or can change dynamically.

Image Processing Agent

- The Image Processing Agent (IPA) uses a simplified NOAA algorithm to perform a cloud cover calculation and reports the results to RM to help establish the benefit of a particular image.
- The NOAA algorithm has been modified to skip pixels in order to speed up the calculation, thus lowering the amount of processor time required.
- If RM determines that there is not enough processor time available, then RM can change the fidelity of the IPA, by setting the proper service attribute such that the NOAA algorithm skips a certain number of pixels.

Compressor Agent

- The Compressor Agent (CA) uses various routines to compress data in order to reduce the buffer space needed by satellite applications.
- If RM determines that the buffer is a constrained resource, then it changes the fidelity of the CA algorithm such that greater compression is performed.

Data Management Sub-System

- The Data Management Agent (DMA) receives a service attribute value from the Meta-Agent telling whether to download or not.
- When the DMA is told to download, the data transfer process will connect to the data receiver, and begin downloading the images in order.

Health and Safety Data Agent

- Produces health and safety information and stores the data as files in the buffer.
- Several metrics are gathered, such as temperature, power, and pressure, at various fidelities.
- The fidelity of gathering a metric determines how often the data is collected, resulting in smaller file sizes when operating at a low fidelity.

Future Work

- Extend satellite *buffer management* to allow greater control by the resource manager.
- Extend benefit framework and specification.
- Extend framework for supporting various types of *extrinsic attributes* and formalize the model of an *extrinsic attribute schedule*.
- Create more advanced *allocation*, *service attribute* and *local optimization algorithms* that are more capable of optimizing the benefit of captured data.
- Advance the *profiling* of application performance.
- *System Certification* – On-line verification that performance requirements will be met.
- Help employ this technology within an actual NASA mission.